2017 8th International Conference on Information Technology (ICIT)

Unified Strategy for Intensification and Diversification Balance in ACO Metaheuristic

Rafid Sagban
Information Networks Dept., College of Information Technology,
University of Babylon, Babylon, Iraq
rsagban@uobabylon.edu.iq
www.uobabylon.edu.iq

Ku Ruhana Ku-Mahamud
School of Computing, College of Arts and Sciences,
Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia
ruhana@uum.edu.my
www.uum.edu.my

Muhamad Shahbani Abu Bakar
School of Computing, College of Arts and Sciences,
Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia
shahbani@uum.edu.my
www.uum.edu.my

***Abstract*—This intensification and diversification in Ant Colony Optimization (ACO) is the search strategy to achieve a trade-off between learning a new search experience (exploration) and earning from the previous experience (exploitation). The automation between the two processes is maintained using reactive search. However, existing works in ACO were limited either to the management of pheromone memory or to the adaptation of few parameters. This paper introduces the reactive ant colony optimization (RACO) strategy that sticks to the reactive way of automation using memory, diversity indication, and parameterization. The performance of RACO is evaluated on the travelling salesman and quadratic assignment problems from TSPLIB and QAPLIB, respectively. Results based on a comparison of relative percentage deviation revealed the superiority of RACO over other well-known metaheuristics algorithms. The output of this study can improve the quality of solutions as exemplified by RACO.**

Keywords*— Metaheuristics; Ant Colony Optimization; Reactive Heuristics; Recursive Local Search; Reward Assignment Strategies

I. INTRODUCTION

In the fields of artificial intelligence and operational research, combinatorial optimization (CO) is aimed to find the “best” solution among a finite set of solutions called *search space* [1]. Examples of CO problems are travelling salesman problem (TSP), quadratic assignment problem (QAP), vehicle routing problem (VRP), and scheduling problem [2]. Due to the practical and theoretical importance of these kinds of problems, several search methods have been proposed to traverse the search space to find the optimal solutions to the given CO problem. Due to the complexity issues resulted from the exponential expanding to the solution search space and in order to avoid premature convergence, alternative stochastic methods called *metaheuristics* have been invented [3]. Metaheuristics sacrificed the long time needed to find the optimal solutions by the quick time to find near-optimal solutions. Talbi [4] classified them according to several criteria explaining their importance, implementation, and performance aspects. Metaheuristics basically are combined heuristic methods in higher-level metaphors. Examples of these metaphors are annealing, memory, evolution, or ant foraging behaviour. The metaheuristics that are inspired from the stated metaphors are simulated annealing (SA), tabu search (TS), evolutionary computation (EC), or ant colony optimization (ACO).

ACO takes the inspiration of its method of search from the foraging behaviour of real ants in nature. Ants, in their continuous journey searching for food, mark chemical paths to be followed by other ant foragers of the colony. This type of indirect communication between ants has been described by the following mathematical model. The model was the result of several experiments known as “double bridge”. The ants, after diversifying the space and finding the food, will return back to the nest using one of the branches of the bridge. By laying and following a chemical substance called pheromone and after time *t*, the ants will find the shortest path between food and nest. These natural optimization processes have been harnessed in the design of the first ant algorithm named ant system (AS) [5]. Even though the original AS algorithm achieved encouraging results for the TSP problem, it was later found to be inferior to the state-of-the-art algorithms for TSP as well as for other CO problems. The unbalanced designs of intensification and diversification mechanisms lead to stagnation problems when all the search agents (i.e. ants) follow the same path. In the purpose of improving the intensification and diversification behaviours, several AS variants have been proposed in the literature review as cited by [6]. The variants form the ACO framework. The substantial difference among AS variants is in the way of guiding the search. It is due to the way of managing the two criteria: exploration of the search space (diversification) and exploitation of the best solutions found (intensification).

Intensification and diversification is the main search strategy that guides the search process. Diversification refers to the process of traversing new regions of search space. Whereas intensification refers to the process of traversing the neighbours of good regions [7]. In other words, diversification concerns the whole search space, while intensification accumulates the previous search experience in some promising regions. The previous experience of search agents is transferred from previous iterations to the next ones. Therefore, at the beginning of the search, the diversity is high because of the randomness in generating the initial solutions (global search); then the agents start to converge towards local optimum points (local search). The initial diversity is proportional to the number of agents involved in finding the initial solutions. After a sufficient number of iterations, the search agents will narrow the search in promising regions. Continuing with intensification leads to the risk of getting trapped in local optima, when all search agents generate the same solutions. The trade-off between intensification and diversification is governed by the way that the intrinsic components are named by the intensification and I&D frame [8].

Reactive search [9] is a framework that integrates machine learning techniques, memory, and online parameters' selection, together with a strategy to increase diversity as needed. Typically, the search is restarted when the premature convergence occurs. A diversity indicator is harnessed as a trigger for restarting the search and as evidence for parameter adaptation. The ability of reactive search as a new technique to maintain the dynamism of the intensification and diversification mechanics entails its integration with ACO to produce a powerful approach for nondeterministic problem-solving. Although reactive search optimization and I&D frame greatly influenced the intensification and diversification automation, existing works in ACO were limited solely to the proper management of pheromone memory or to the adaptation of a few parameters [6]. Therefore, a comprehensive literature review of how much the reactive search and I&D frame principles are applied in the search strategies of ACO is very much needed.

The present study conducted an extensive survey to provide an intuitive and profound understanding of the current situation of ACO research. In addition, this study introduces the reactive ant colony optimization (RACO) approach as a unified search strategy adhering to the main principles of reactive search. Intensification and diversification in ACO is described in Section 2, while the proposed RACO is presented in Section 3. Conclusions are presented in Section 4.

II. DIVERSIFICATION AND INTENSIFICATION IN ACO

The widespread attention that ACO has received made it more popular than other metaheuristics [10]. However, it suffers, just like other search algorithms, a problem of reaching the global optima in its found solutions. The problem becomes progressively worse as the search space increases [11]. To achieve a good balance between intensifying and diversifying the search space, the orientation of the recent ACO literature is five-fold.

Firstly, the ACO literature tends to adapt the entire pheromone update and construct solution strategies to reduce the risk of getting trapped in the search stagnation problem [12] [13] [14]. Secondly, a literature trend [15] has discussed several intensification/diversification strategies only from a specific point of view, such as considering one combinatorial problem. Thirdly, the ACO diversification is combined with intensification of local search strategies [16] [17]. Fourthly, orientation [18] [19] advocates the hybridization with other population-based methods (rather than local search). Fifthly, orientation [20] tends to hybridize ACO with more than one local search and population-based method in an excessive and miss-leaded way. This paper's argument is that using unclear methodologies could cause a chaotic search strategy and will lose the dynamism control of intensification and diversification.

III. PROPOSED REACTIVE ANT COLONY OPTIMIZATION

In RACO, the trade-off between intensification and diversification is automated based on the reactive search optimization. It has been built based on the reactive model proposed by Khichane et al. [21] and the ACO model proposed by Stützle and Hoos [22]. The automation is maintained by addressing the main I&D problems of memorization, diversity indication, and parameterization in the said models. The general scheme of RACO is presented in Figure 1.

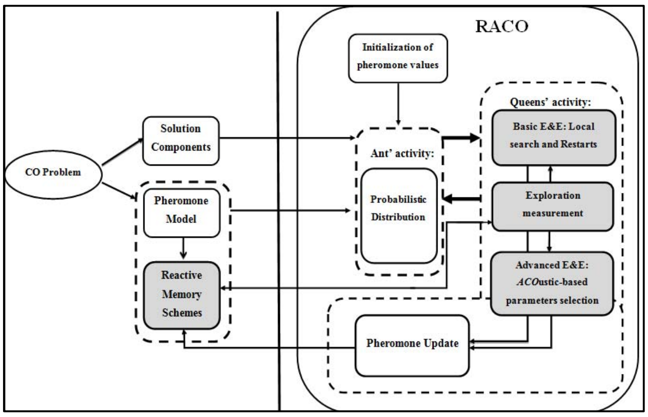

Fig. 1. The general scheme of RACO

RACO starts solving CO problems by iterating two activities, namely ants' activity and queen's activity. An example of the ants' activity is the probabilistic solution construction, where each ant is able to take individual decisions. Whereas, an example of the queen's activity is every central decision that can be taken to change the current search status. CO problems (such as TSP and QAP) are assembled as a finite set of solution components. Next, a set of pheromone values called the pheromone model is defined. The set of pheromone values is parameterized probabilistically to be used then in generating solutions based on the solution components. Two reactive memory schemes are defined. The first scheme is associated with the neighbourhood structure formed by ants, while the second scheme is associated with the neighbourhood structure formed by local search procedures. The candidate solutions are constructed using the pheromone model. The pheromone values are updated by the queen in such a way that it is biased in future towards high quality solutions.

For this part of RACO, there are two basic I&D mechanisms: the reactive restart mechanism and the recursive local search (RLS) mechanism. In the former mechanism, the neighbourhood drawn by ants is traversed before the restart just to record the unpromising regions. The regions are simply characterized using τ_{min} threshold, where the components of solutions below this threshold will be recorded in the first scheme in terms of reactive heuristics (RH). After the restart, the reactive heuristics will be used as a guidance for the ants to decide the next component in the constructed solution. The exploration measure is the I&D component that decides the proper point of restart, which is when the search is stagnated. RACO uses an *ACO*ustic indicator [23] to characterize the diversity of search. Based on this indication, RACO can decide whether the search is over-explorative or over-exploitative. When the first situation is active, a RLS mechanism [24] takes control. In RLS, an *old-best-so-far* solution will be recorded in the second memory scheme to be used in next iterations as a reference for improvement in the quality of solutions. If the new produced solution is better than the *old-best-so-far* solution, it will be recorded in the memory; otherwise, the *old-best-so-far* solution will be recorded again in the memory. In contrast, when the search is over-exploitative, the search will be restarted and reactive heuristics [25] will be active to induce the ants to explore untraversed regions. Using the *ACO*ustic indicator, the queen in this way controls the reinforcement learning process inside the colony by forcing other ants for being exploitative agents or explorative agents. In the former choice, they keep searching around the structure of the neighbourhood of good solutions, whereas, in the latter choice, they shift the search to another neighbourhood structure. The last part of the queen's activities is the APS_{ACO} mechanism [26], in which the queen controls the way of the search based on the feedback collected from the search process. An internal reinforcement learning process is involved to learn the parameter values during the run.

The adaptation of APS_{ACO} in RACO is the last step in automating intensification/diversification in an ACO-based reactive search. Three variants of APS_{ACO} algorithm can be alternated by varying the proposal of the perspective strategy. These are quality-based (QRA), exploration-based (ERA), and unified-based (URA) reward strategies. In RACO, the general improvement in the quality of solutions (QRA) uses a proxy for the impact of the selected parameters. Finally, RACO is evaluated against other similar metaheuristics.

A. Experimental Design

The performance of RACO is evaluated by the comparison with other metaheuristic approaches to solve TSP and QAP. The evaluation metric is reported using the relative percentage deviation (RPD) metric as follows:

$$RPD = \frac{\text{the result cost} - \text{the best known solution}}{\text{the best known solution}} * 100 \quad (1)$$

The maximum number of iterations is equal to the same number of tours for the algorithms with which RACO is compared. An average of ten trails for the results is reported. For the RACO parameter settings, the neighbourhood threshold is fixed to (0.8) without tuning. The number of ants (m) is equal to (5), while the rest of the RACO parameters are configured adaptively using the QRA strategy. Hence, the RACO variant used in the experiments is denoted as $RACO_{QRA}$. For TSP, the instances are taken from TSPLIB [27]; they are *Burma14*, *Dantzig42*, *Oliver30*, *Eil51*, *Eil76*, *KroA100*, and *Eil101*. The configuration of experiments is dictated based on the availability of the published results. Numerical experiments are executed to regenerate the results of other algorithms; otherwise, their performance is taken from the literature. The results of ant colony system (ACS) are based on the implementation included in ACOTSP.V1.3 [28].

Other algorithms with which RACO is compared to are SA, evolutionary programming (EP), genetic algorithm (GA), particle swarm optimization (PSO), and artificial bee colony (ABC). The results of SA and EP are from Dorigo and Gambardella (1997). The results of GA and PSO are from Çunkaş and Özsağlam (2009), and the results of ABC are from Kocer and Akca (2014). For QAP, the benchmark data is taken from QAPLIB [32], namely *Nug30*, *Ste36b*, *Tai30a*, *Tai40a*, *Tai50a*, *Tai60a*, *Tai80a*, and *Tai100a* for the random-generated category. The results of the algorithms used in the comparison are taken from the literature. The performance of object-guided ant colony optimization (OG-ACO) and hybrid artificial fish-school optimization (HAFSO) are from [33]. The run length is based on [33].

B. Results

A series of experiments to evaluate the performance of RACO, in terms of relative percentage deviation from the well-known solutions for TSP and QAP, has been conducted. In terms of TSP, based on the experiment (Figure 2), the proposed algorithm achieved a 100% success rate by reaching the known optimum at the first four turns. The rates were 99% and 92% for the fifth and sixth turns. It was observed that RACO ended with 0% margin of error in small-scaled TSP problems. The results confirmed that the combination of RH, RLS technique, and QRA controller produces high quality solutions.

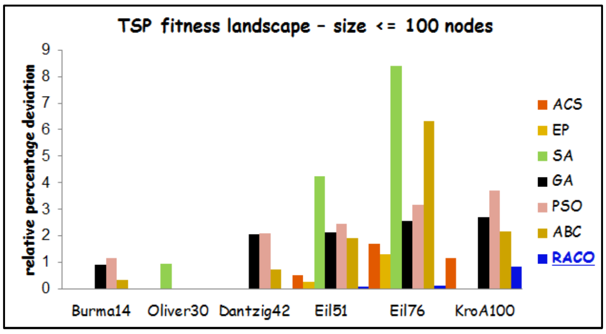

Fig. 2. Results of comparing RACO with ACS, EP, SA, GA, PSO, and ABC algorithms in small-scaled tsp instances using RPD test

It is observed that RACO performs well in terms of solution qualities and diversification dynamics. It is suspected that with large instances, the importance of reactive heuristics becomes less due to the lower number of restart triggers. Observations from the three experiments suggest exchanging the traditional triggers (such as λ-branching factor) with machine learning triggers (such as *ACO*ustic [23]).

In terms of QAP, RACO is applied to random-generated instances. The results of the experiments are reported. The results confirmed that the quality of solutions produced by the RACO algorithm is better than others for QAP. From Figure 3, the results of the experiments on the random-generated instances show that RACO is better than other methods with all the scales of this type of QAP instances. The superiority of RACO to the modern swarm intelligence methods, i.e. OG-ACO and HAFSOA, confirms the harmonic combination of I&D components of RACO (Figure 3). The advantage of diversified components, such as reactive heuristics, enables RACO to avoid early convergence. The advantage of intensified components, i.e. recursive local search together with maintaining I&D automation using a parameter control, enables RACO to sustain the search around promising regions, which improves the quality of solutions.

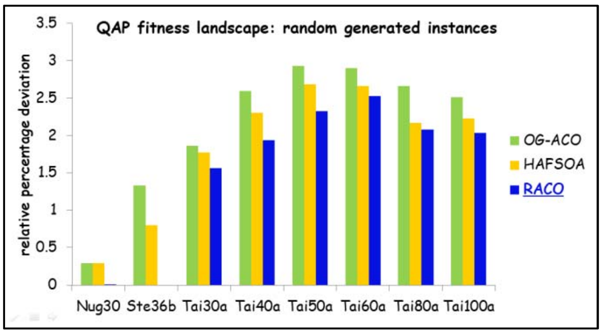

Fig. 3. Results of comparing RACO with OG-ACO and HAFSOA algorithms in random generated QAP instances using RPD test

CONCLUSION

Intensification and diversification is the most important strategy of search in ant colony optimization. The schemes are responsible for managing the neighbourhood structures drawn by ants using the pheromone memory model and for managing the neighbourhood structures drawn by local search procedures. The outcome of this paper is to provide a fresh treatment of intensification and diversification balance automation by RACO, the reactive ant colony optimization. In RACO, there is a harmonic combination of I&D memory-based schemes, robust diversity indication using quality and/or diversity metrics, and generic parameter controller based on the said metrics. RACO is validated using the benchmark data from TSPLIB and QAPLIB. The results were encouraging in terms of the quality of solutions.

ACKNOWLEDGMENT

The authors wish to thank Universiti Utara Malaysia in funding this study under The Postgraduate Incentive Grant (S/O code 15528) and AHSGS for the administration of this study. The authors are very grateful to the anonymous reviewers for their valuable suggestions and comments to improve the quality of this paper.

REFERENCES

[1] R. Fletcher, *Practical methods of optimization*. Chichester, England: Weley & Sons Ltd., 1997.

[2] D. Bertsimas, D. B. D. Brown, and C. Caramanis, “Theory and Applications of Robust Optimization,” *SIAM Rev.*, vol. 53, no. 3, pp. 464–501, 2011.

[3] M. Gendreau and J.-Y. Potvin, *Handbook in Metaheuristics*. New York, USA: Springer, 2010.

[4] E.-G. Talbi, *Metaheuristics from design to implementation*. U.S.A.: Wiley, 2009.

[5] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system:

Optimization by a colony of cooperating agents," *IEEE Trans. Syst.*, vol. 26, no. 1, pp. 29–41, 1996.

[6] M. Dorigo and T. Stützle, "Ant colony optimization: Overview and recent advances," in *Handbook of metaheuristics*, vol. 146, M. Gendreau and J. Potvin, Eds. New York, USA: Springer US, 2010, pp. 227–263.

[7] M. Crepinsek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms : A survey," *ACM Comput. Surv.*, vol. 45, no. 3, pp. 1–33, 2013.

[8] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, 2003.

[9] R. Battiti, M. Brunato, and F. Mascia, *Reactive search and intelligent optimization*. U.S.A.: Springer, 2008.

[10] A. Prakasam and N. Savarimuthu, "Metaheuristic algorithms and probabilistic behaviour: a comprehensive analysis of Ant Colony Optimization and its variants," *Artif. Intell. Rev.*, vol. 45, no. 1, pp. 97–130, 2016.

[11] M. Yoshikawa and T. Nagura, "Adaptive Ant Colony Optimization Considering Intensification and Diversification," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2009, vol. I, no. 1, pp. 200–203.

[12] F. Olivas, F. Valdez, and O. Castillo, "Dynamic parameter adaptation in Ant Colony Optimization using a fuzzy system for TSP problems," in *16th World Congress of the International Fuzzy Systems Association (IFSA) and the 9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT) Dynamic*, 2015, pp. 765–770.

[13] T. Satukitchai and K. Jeranaitanakij, "An Early Exploratory Method to Avoid Local Minima in Ant Colony System," in *2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2015, no. 1, pp. 1–5.

[14] Y. Yan, "A Modified ant system to acheive better balance between intensification and diversification for the traveling salseman problem," Unpublished doctoral dissertation. New Mexico State University, Las Cruces, New Mexico, 2016.

[15] C. B. Kalayci and C. Kaya, "An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery," *Expert Syst. Appl.*, vol. 66, no. 2016, pp. 163–175, 2016.

[16] J. Guan and G. Lin, "Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem," *Eur. J. Oper. Res.*, vol. 248, no. 3, pp. 899–909, 2016.

[17] R. M'Hallah and A. Alhajraf, "Ant colony systems for the single-machine total weighted earliness tardiness scheduling problem," *J. Sched.*, vol. 19, no. 2, pp. 191–205, 2016.

[18] M. Yousefikhoshbakht, N. Malekzadeh, and M. Sedighpour, "Solving the Traveling Salesman Problem Based on The Genetic Reactive Bone Route Algorithm whit Ant Colony System," *Int. J. Prod. Manag. Eng.*, vol. 4, no. 2, p. 65, 2016.

[19] A. Fouad, D. Boukhetala, F. Boudjema, and K. Zenger, "A novel global Harmony Search method based on Ant Colony Optimisation algorithm - Journal of Experim," *J. Exp. Theor. Artif. Intell.*, vol. 28, no. 1, pp. 215–238, 2016.

[20] A. M. Mohsen, "Annealing Ant Colony Optimization with Mutation Operator for Solving TSP," *Comput. Intell. Neurosci.*, vol. 2016, p. 13, 2016.

[21] M. Khichane, P. Albert, and C. Solnon, "An ACO-based reactive framework for ant colony optimization: First experiments on constraint satisfaction problems," in *Lecture notes in computer science (LNCS): Learning and intelligent optimization*, vol. 5851, T. Stützle, Ed. Springer Berlin Heidelberg, 2009, pp. 119–133.

[22] T. Stützle and H. H. Hoos, "MAX – MIN ant system," *Futur. Gener. Comput. Syst.*, vol. 16, no. 8, pp. 889–914, 2000.

[23] R. Sagban, K. R. Ku-Mahamud, and M. S. A. Bakar, "ACOustic : A Nature-Inspired exploration indicator for ant colony optimization," *Sci. world J.*, vol. 2015, pp. 1–11, 2015.

[24] R. Sagban, K. R. Ku-mahamud, and M. S. A. Bakar, "Reactive max-min ant system with recursive local search and its application to tsp and qap," *Intell. Autom. Soft Comput.*, vol. 23, no. 1, pp. 127–134, 2017.

[25] R. Sagban, K. R. Ku-Mahamud, and M. S. A. Bakar, "Reactive memory model for ant colony optimization and its application to TSP," in *Proceedings of ICCSCE2014: The 4th International Conference on Control System, Computing and Engineering*, 2014, pp. 310–315.

[26] R. Sagban, K. R. Ku-Mahamud, and M. S. A. Bakar, "Nature-Inspired Parameter Controllers for ACO-based Reactive Search," *Res. J. Appl. Sci. Eng. Technol.*, vol. 11, no. 1, 2015.

[27] G. Reinelt, "Benchmark-TSPLIB:A traveling salesman problem library," *ORSA Journal On Computing*, 1991. [Online]. Available: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.

[28] T. Stützle, "Software-ACOTSP, Version 1.03.," 2004. [Online]. Available: http://www.aco-metaheuristic.org/aco-code.

[29] M. Dorigo and L. M. Gambardella, "Ant colony system : A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, 1997.

[30] M. Çunkaş and M. Y. Özsağlam, "A Comparative Study on Particle Swarm Optimization and Genetic Algorithms for Traveling Salesman Problems," *Cybern. Syst.*, vol. 40, no. 6, pp. 490–507, 2009.

[31] H. E. Kocer and M. R. Akca, "an Improved Artificial Bee Colony Algorithm With Local Search for Traveling Salesman Problem," *Cybern. Syst.*, vol. 45, no. 8, pp. 635–649, 2014.

[32] R. . Burkard, E. Cela, S. E. Karisch, and F. Rendl, "Benchmark-QAPLIB :A Quadratic Assignment Problem Library," *J. Glob. Optim.*, no. 10, pp. 391–403, 1997.

[33] L. Ziqiang and Y. Yi, "A hybrid artificial fish-school optimization algorithm for solving the quadratic assignment problem," in *2014 10th International Conference on Natural Computation (ICNC)*, 2014, no. 61272294, pp. 1099–1104.